

# What's new in Page Forms and the rest

**Yaron Koren • SMWCon Fall 2022 • October 26, 2022**

# About me

- "Enterprise MediaWiki" developer, consultant and evangelist, based in New Jersey, USA
- My extensions: Page Forms, Cargo, External Data, Replace Text, Approved Revs, etc.
  - See also: [Canasta!](#)
- Founder/CEO of [WikiWorks](#), now also the CTO of Wikiask ([wikiask.org](http://wikiask.org))
- My book: *[Working with MediaWiki](#)*
- My podcast: *[Between the Brackets](#)*

# **What's new in Page Forms**

**("new" = added in the last year)**

# Values from Wikidata

- In a "Building" form definition:

```
'''Country''': {{{field|Country|input type=combobox  
|values from wikidata=P31=Q6256}}}
```

("Instance of" = "Country")

- "Dependent autocomplete" is also supported:

```
'''City''': {{{field|City|input type=combobox|values  
from wikidata=P31=Q515&P17=Building[Country]}}}
```

("Instance of" = "City", "Country" = a previous form field)

# Why Wikidata specifically?

Page Forms already supports getting form values from outside data sources (using values from external data) – so why special handling for Wikidata?

- values from wikidata, unlike values from external data, runs another query every time the user types a character ("remote autocompletion"), supporting much larger data sets.
- Plus, Wikidata is the most important data source there is!

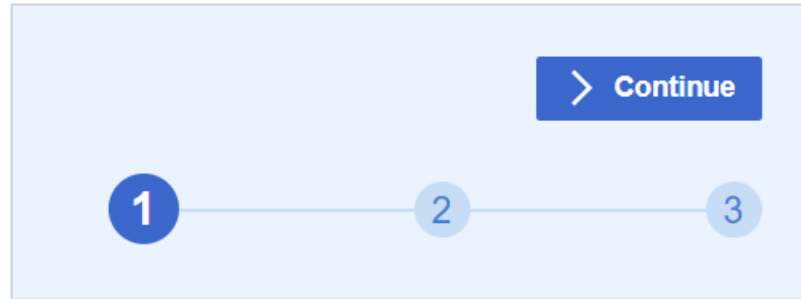


# Wizard-style display - appearance

Field 1:

Field 2:

Field 3:



A light blue rectangular box containing a progress indicator and a navigation button. The progress indicator consists of three circles connected by a horizontal line. The first circle is dark blue with a white number '1'. The second and third circles are light blue with dark blue numbers '2' and '3' respectively. In the top right corner of the box is a dark blue button with a white right-pointing chevron and the text 'Continue'.

(The third screen in this form holds the standard inputs – the "Save" button, etc.)

# "category" option in #template\_params

```
{{#template_params:  
...  
|Project manager (category=Person)  
...  
}}
```

This affects:

- Editing - automatic "values from category" in the form field
- Display – red links get linked to the form for that category



# **What's new in Page Schemas**

# Better display

**Template**




Name:


☐ Allow multiple instances of this template


Output format: ☒ Table ☐ Side infobox ☐ Plain text ☐ Sections


☒ **Cargo**

Table name:

 **Field: Organization**  

 **Field: Email address**

 **Field: Phone number**

 **Add field**

Field <div>s are now automatically minimized and sortable, very similar to the handling of "multiple-instance templates" within Page Forms.

# **What's new in External Data**

# External Data

- An extension that allows for querying outside data sources (APIs, databases, files, etc.) and displaying their data on the wiki page
- Huge amount of additions/restructuring of the extension in the last three years, mostly done by Alexander Mashin (who is now a co-author)
- I will just cover some additions from the last year.

# Streamlined settings

- External Data formerly had various \$edg... global settings
- These are now deprecated, replaced by \$wgExternalDataSources – a single global array that can hold many settings per data source.

# Fewer required params

- "data" now sometimes optional
  - If it's just `data=Name=Name,Population=Population,...`, it can just be removed
- "format" now (usually) optional
  - Auto-detection of formats like CSV, XML, etc.

# New supported formats

- INI ("key=value" pairs, one per line)
- YAML (an extension of JSON)

(These are in addition to the existing CSV, JSON, XML, etc.)

# File wildcards

```
{{#get_file_data:  
source=LabStudyResultsDirectory  
| file=*.csv  
| format=csv with header  
}}
```

This will get the data from all CSV files within the directory, and put it all into one big table.



# Reading archive files

- Online or local files of the formats .zip, .rar, .tar, .tar.gz and .tar.bz2 can now be parsed.

Example:

```
{{#get_file_data:source=MyLocalZipFile  
| archive path=/hr/employees/EmployeeData.csv  
| format=csv  
| data=Name=Name,Department=Dept,Title=Title}}
```

- "archive path", too, can hold wildcards

# #format\_external\_table

- Displays a table of information, like #for\_external\_table, but uses Cargo result formats to do it.

Example:

```
{{#format_external_table:db=MyDB|from=Museums  
|data=name=museum_name,address=museum_address  
|hours=museum_hours|display format=dynamic table}}
```

- Cargo is required for this feature.

# #get\_inline\_data

- Retrieves information from structured text contained directly within the call.

Example:

```
{{#get_inline_data:  
text=<AvgRainfall>23.5</AvgRainfall>  
|format=xml|data=rainfall=AvgRainfall}}
```

# Integrated data retrieval and display

- "Old"/standard way to use External Data (AKA "legacy mode"):

```
{{#get_web_data:url=https://myapis.com/ClimateInfo?country=Indonesia|format=json|data=rainfall=rainfall}}
```

The average rainfall of Indonesia is

```
{{#external_value:rainfall}}.
```

- This might not work when MediaWiki switches to Parsoid, because the order of calling will not be guaranteed (`#external_value` might get called before `#get_web_data`).

# Integrated retrieval and display (cont.)

- New way to use External Data (AKA "standalone mode"):

```
The average rainfall of Indonesia is  
{{#external_value:rainfall  
|url=https://myapis.com/ClimateInfo?country=Indonesia  
|format=json|data=rainfall=rainfall}}.
```

- Eventually, the parser functions #get\_web\_data, #get\_db\_data, etc. (including #get\_inline\_data!) will all be removed.

# Future of External Data

Eventually, it may make sense to reduce External Data down to just one parser function (and one corresponding Lua method).

- Data source type specified in LocalSettings.php (unless it's a URL or plain text)
- Standard querying params ("order by=", "where=", "limit=", etc.) that work across data source types
- Display method determined by which parameter is specified: "display text=", "display template=" or "display format="

# Future of External Data (cont.)

Possible call:

```
{{#external_data  
|source=ContractsDataSource  
|where=Client=Acme N.V.  
|order by=Date  
|display format=table  
}}
```

(What's the source type? Could be anything!)

# **What's new in Flex Diagrams**



# Handling for diagrams.net (AKA Drawio)

diagrams.net (formerly known as Drawio) is a popular open source JS diagramming tool.

This joins the other already-supported diagram types:

- BPMN (using bpmn-js)
- Gantt (using DHTMLX)
- Mermaid

**Thank you!**

**Questions/comments**